Assignment-6 ( Python Basics)

Q1. What are keywords in python? Using the keyword library, print all the python keywords.

Solution:

Python keywords are special reserved words that have specific meanings and purposes and can't be used as variable or function name. These keywords are always available, no need for import them into the code.

import keyword

print(keyword.kwlist)

output:

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Q.2. What are the rules to create variables in python?

* A variable name must start with a letter or the underscore character.
* A variable name cannot start with a number.
* A variable name can only contain alpha-numeric characters and underscores.
* Variable names are case-sensitive (hello, Hello and HELLO are three different variables)

Q.3. What are the standards and conventions followed for the nomenclature of variables in

python to improve code readability and maintainability?

Solution:

*1.pothole\_case\_naming* uses lowercase words separated by underscores \_. This is our suggested format as the underscores make it easy to read the variable, and don’t add too much to the length of the variable name. As an example, consider the variable temp\_celsius. In the context of the examples above, we might rename our variable fmi\_station\_id, which conveys all of the essential information we need, while remaining easy to read.

*2.camelCase* or *CamelCase* uses capitalization of the first letter of words in a variable name to make it easier to read. In some cases the first letter of the variable may be capitalized. The variable tempFahrenheit was one example of camelCase. Again, if we consider the examples from the previous section, we might consider the variable name fmiStationID or simply stationID if we elect to use camelCase.

Q.4. What will happen if a keyword is used as a variable name?

Solution:

Keywords are reserved words. we cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language. It will produce error.

Q.5. For what purpose def keyword is used?

The def keyword is used to create, (or define) a function.

Q.6. What is the operation of this special character ‘\’?

Solution:

Blackslash which is used to separate locations in the file or network path.

Q.7. Give an example of the following conditions:

Solution:

(i) Homogeneous list

[2,34,5,11,26,8]

(ii) Heterogeneous set

{‘hello’,23,5,78,4,4,3,28,5,’welcome’}

(iii) Homogeneous tuple

(‘krish’,’Anand’,’sweety’,’Rakesh’)

Q.8. Explain the mutable and immutable data types with proper explanation & examples.

Solution:

|  |  |
| --- | --- |
| **Mutable Objects** | **Immutable Objects** |
| A mutable object can be changed after it is created | An immutable object cannot be changed after it is created |
| Examples : List, Set, Dictionary | Example: tuples, int, float, bool |
| Mutable objects are not considered as thread-safe in nature | Immutable objects are regarded as thread-safe in nature |
| Mutable Objects are slower to access, as compared to immutable objects | Immutable objects are faster to access when compared to mutable objects |
| Mutable objects are useful when we need to change the size or contents of our object. | Immutable objects are best suitable when we are sure that we don't need to change them at any point in time. |
| Changing mutable objects is a cheaper operation in terms of space and time | Changing immutable objects is an expensive operation since it involves creating a new copy for any changes made. |

Q.9. Write a code to create the given structure using only for loop.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

Solution:

initial\_star=1

for i in range(initial\_star,10,2):

print('\*'\*i)

Q.10. Write a code to create the given structure using while loop.

|||||||||

|||||||

|||||

|||

|

Solution:

initial =0

while initial<9:

no\_of\_times=9-initial

print('|'\*no\_of\_times)

initial+=2